

Cheat Sheet for comprehensive Sass

Variables

- **Definition:** Store reusable values like colors, fonts, and sizes.

```
$primary-color: #3498db;
$font-stack: Helvetica, sans-serif;
```

- **Usage:** Reference variables in styles.

```
body {
  color: $primary-color;
  font-family: $font-stack;
}
```

Nesting

- **Basic Nesting:** Nest selectors to reflect HTML structure.

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li { display: inline-block; }
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

- **Parent Selector (&):** Reference the parent selector.

```
a {
  color: blue;
  &:hover { color: red; }
}
```

Mixins

- **Definition:** Reusable blocks of styles.

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  border-radius: $radius;  
}
```

- **Usage:** Include mixins in styles.

```
.button {  
  @include border-radius(10px);  
}
```

Functions

- **Definition:** Custom functions to compute values.

```
@function calculate-width($width) {  
  @return $width * 2;  
}
```

- **Usage:** Use functions to compute values.

```
.container {  
  width: calculate-width(100px);  
}
```

Extend/Inheritance

- **Extend:** Share styles among selectors.

```
%message-shared {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.message {  
  @extend %message-shared;  
}  
  
.success {
```

```
@extend %message-shared;
border-color: green;
}
```

Partials and Imports

- **Partials:** Organize code into smaller files.

```
// _base.scss
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

- **Imports:** Include partials in main file.

```
// main.scss
@import 'base';

.container {
  width: 100%;
  overflow: hidden;
}
```

Operators

- **Arithmetic Operators:** Perform calculations.

```
.container {
  width: 100% / 3;
}
```

- **Comparison Operators:** Compare values.

```
@if $var > 10 {
  .element { color: red; }
}
```

Control Directives

- **If/Else:** Conditional statements.

```
@mixin text-style($weight) {
  @if $weight == bold {
    font-weight: bold;
  } @else {
    font-weight: normal;
  }
}
```

- **For Loop:** Iterate over a range.

```
@for $i from 1 through 3 {
  .item-#{ $i } { width: 2em * $i; }
}
```

- **Each Loop:** Iterate over a list.

```
@each $color in red, green, blue {
  .#{ $color }-bg { background-color: $color; }
}
```

Maps

- **Definition:** Key-value pairs.

```
$colors: (
  primary: #3498db,
  secondary: #2ecc71,
  danger: #e74c3c
);
```

- **Usage:** Access map values.

```
.element {
  color: map-get($colors, primary);
}
```

Placeholder Selectors

- **Definition:** Placeholder selectors for extend.

```
%button-base {
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}
```

- **Usage:** Extend placeholders.

```
.button {
  @extend %button-base;
  background-color: #3498db;
}
```

Color Functions

- **Manipulation:** Adjust colors.

```
$primary-color: #3498db;
.element {
  background-color: lighten($primary-color, 10%);
  border-color: darken($primary-color, 10%);
}
```

Media Queries

- **Nesting:** Nest media queries within selectors.

```
.container {
  width: 100%;
  @media (min-width: 600px) {
    width: 50%;
  }
}
```

Comments

- **Single-line:** `// Comment`
- **Multi-line:** `/* Comment */`

Advanced Tips

- **Debugging:** Use `@debug` to print values to the console.

```
@debug 10em + 12em;
```

- **Source Maps:** Enable source maps for easier debugging.

```
sass --sourcemap=inline input.scss output.css
```

- **Custom Functions:** Extend Sass with custom functions.

```
@function str-replace($string, $search, $replace: '') {
  $index: str-index($string, $search);
  @if $index {
    @return str-slice($string, 1, $index - 1) + $replace + str-
replace(str-slice($string, $index + str-length($search)), $search,
$replace);
  }
  @return $string;
}
```

Best Practices

- **Modular Code:** Use partials and imports for modularity.
- **DRY Principle:** Avoid repetition with mixins, functions, and extend.
- **Consistency:** Maintain consistent naming conventions and structure.
- **Documentation:** Comment your code for clarity and maintainability.

Common Pitfalls

- **Over-nesting:** Avoid excessive nesting to prevent overly specific selectors.
- **Performance:** Be mindful of the size and complexity of your compiled CSS.
- **Browser Compatibility:** Ensure mixins and functions handle vendor prefixes appropriately.

Resources

- **Official Documentation:** [Sass Documentation](<https://sass-lang.com/documentation>)
- **Community Forums:** [Sass Stack Overflow](<https://stackoverflow.com/questions/tagged/sass>)
- **Libraries:** [Compass](<http://compass-style.org/>), [Bourbon](<https://www.bourbon.io/>)

By Ahmed Baheeg Khorshid

ver 1.0